Some Aspects of Evolutionary Designing Optimal Controllers

Jacek Szczypta, Andrzej Przybył, and Krzysztof Cpałka

Częstochowa University of Technology,
Institute of Computational Intelligence, Poland
{jacek.szczypta,andrzej.przybyl,krzysztof.cpalka}@iisi.pcz.pl

Abstract. In this paper a new automatic method of control system design was presented. Our method is based on the evolutionary algorithm, which is used for selection of the controller structure as well as for parameters tuning. This is realized by means of testing different controller structures and elimination of spare elements, taking into account theirs impact on control quality factors. Presented method was tested with two control objects of different complexity.

1 Introduction

Automatic control is an important issue from scientific and practical point of view (see e.g. [24]). It has a significant impact on the quality and efficiency of industrial processes and human safety. It should be noted that in many practical cases the automatic control systems do not have the optimal structure or parameters. This is due to difficult and time-consuming process of selecting the optimal structure and parameters of the actual control system. Commonly used design process of control systems usually relies on the knowledge and experience of experts. Design process also uses simplified (i.e. usually linear) model of controlled objects. The selection of the optimal control system for real control object (different from the simplified model) must be carried out by trial and error. In general, every admissible control structure should be tested and on this basis chosen is the best one.

In the scientific literature much attention was devoted to the controller design issues. There are, among other ideas, described model-based controllers - i.e. which need the model and parameters of the controlled object (for example the model reference adaptive controllers (MRAC) see e.g. [25]), controllers which do not need the object model - like PID controllers (see e.g. [19], [20]) and sliding mode controllers (see e.g. [5]). In the last years there were widely used a nonlinear controllers which are based on soft-computing techniques (see e.g. [10], [12], [17]). Some authors use soft-computing techniques in synthesis of controllers structure (see e.g. [14]).

The controller that best meets the specific requirements of a given control system is chosen from a set of various controller structures of different properties. The choice is done by the human expert, based on his knowledge and experience.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 91–100, 2013.
© Springer-Verlag Berlin Heidelberg 2013

In comparison with other methods available in the literature, the method presented in this paper allows for both controller structure selection and parameter tuning. This is done automatically using an evolutionary algorithm and based on the accurate model of the controlled object and the controller (see e.g. [1]). As a result, the optimal, i.e. maximizing of the value of used performance index, controller is designed.

This paper is organized into five sections. In Section 2 we describe an idea of the new method for designing optimal controllers. In Section 3 we present a detailed description of the new method for designing optimal controllers. In Section 4 simulation results are presented. Conclusions are drawn in Section 5.

2 Idea of the New Method for Designing Optimal Controllers

The method presented in this paper is dedicated for both controller structure selection and parameter tuning (Fig. 1). This is realized automatically using evolutionary algorithm and based on the as far as possible accurate controller and model of the controlled object. The idea of the method is that the evolutionary algorithm uses the simulation model with the constraints and nonlinearities of the actuators, the controlled object, measurement errors etc. As a result of the evolutionary algorithm activity, the controller is well suited to work with real conditions. This is not possible with the use of any other (i.e. analytical) tuning methods (see e.g. [18], [22], [23]).

The important part of the presented method is the ability to select the optimal controller structure. The optimal structure selection occurs, because during evolutionary learning some controller parts may be eliminated (see e.g. [2], [4]). Of course, elimination does not have a significant negative impact on control quality.

The aim of the evolutionary algorithm is to maximize properly defined fitness function. Its value depends on a few control quality indicators such as: root mean square error (RMSE) value (i.e. difference between required and current control signals), controller structure complexity, existence of the overshoot of the control signals, total harmonic distortion (THD) of the control signals etc. Thus, the obtained controller is optimal, i.e. it maximizes the value of used performance index. Fitness function evaluation involves testing of controller, with its complex structure and appropriate parameters, on a realistic model of the whole control system. Fitness function will be described in the next part of this paper.

Steps of the method used in this paper are the same as in typical evolutionary algorithm (see e.g. [3], [6] - [9], [11], [13], [15], [21]), and are as follows:

- **Step 1.** Initialize chromosome population. Every chromosome codes a single structure of the controller and its parameters.
- **Step 2.** Chromosome population evaluation. Aim of this step is evaluation of the control systems coded in the population of the chromosomes in the sense of adopted criterion.

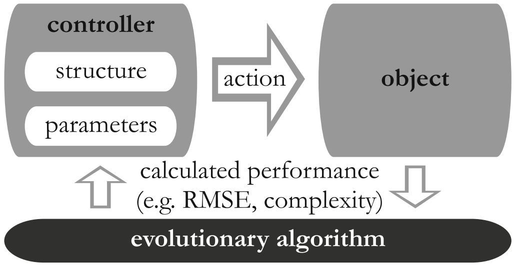

Fig. 1. Idea of the new method for optimal control structure and parameter set selection

- **Step 3.** Evolutionary algorithm stop condition checking. When the best chromosome in population (coding information about best control system in the sense of adopted control criterion) satisfies the stop condition, the algorithm returns information about this chromosome and exits. Otherwise, the algorithm goes to step 4.
- **Step 4.** Chromosome selection for evolutionary operations.
- **Step 5.** Crossover and mutation operators application. This step includes also repair of chromosomes that were obtained from evolutionary operations. Aim of the repair is to correct values of the genes, that are coding control system parameters, to preserve in acceptable range.
- **Step 6.** Generate offspring chromosome population and then go to step 2.

In section 3 we present a detailed description of the new method for designing optimal controllers.

3 Detailed Description of the New Method for Designing Optimal Controllers

Detailed information aboutthe proposed in this paper method of optimal controller design can be summarized as follows:

- **Initialization.** Parent chromosome population initialization includes random generation of genes values. Values are taken from search range (see e.g. [7], [8]). Search range should be customized individually for every single problem.
- **Coding.** Every chromosome $\mathbf{X}_{ch}^{\mathrm{par}}$ codes full parameter set of control system. In this paper the controller composed of typical PID controllers (see e.g. [16]) is considered (Fig. 2). Every gene of the chromosome codes single real value of some controller parameter. Chromosome $\mathbf{X}_{ch}^{\mathrm{par}}$ is described as follows:

$$\mathbf{X}_{ch}^{\mathrm{par}} = (P_1, I_1, D_1, P_2, I_2, D_2, \ldots) = \left(X_{ch,1}^{\mathrm{par}}, X_{ch,2}^{\mathrm{par}}, \ldots, X_{ch,L}^{\mathrm{par}}\right), \quad (1)$$

where P_1, I_1, D_1,..., denote control system parameter values, $ch = 1, .., Ch$, Ch denotes a number of chromosomes in the population, L denotes length of

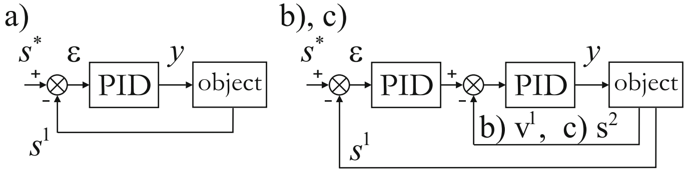

Fig. 2. Control systems structures tested in the experiments: a) 1xPID-s b) 2xPID-v c) 2xPID-s

the chromosome $\mathbf{X}_{ch}^{\text{par}}$. It is important to notice that, for each chromosome $\mathbf{X}_{ch}^{\text{par}}$ in population there is chromosome $\mathbf{X}_{ch}^{\text{red}}$. Binary genes of the chromosome $\mathbf{X}_{ch}^{\text{red}}$ decide which parts of elementary PID controllers should occur in control system. Chromosome $\mathbf{X}_{ch}^{\text{red}}$ is described as follows:

$$\mathbf{X}_{ch}^{\text{red}} = \left(X_{ch,1}^{\text{red}}, X_{ch,2}^{\text{red}}, \ldots, X_{ch,L}^{\text{red}}\right), \tag{2}$$

where every gene in $X_{ch,g}^{\text{red}} \in \{0,1\}$, $ch = 1,..,Ch$, $g = 1,..,L$, decides, if relevant part of control system occurs in control process (relevant gene $X_{ch,g}^{\text{red}} = 1$). It should be noted, that chromosome $\mathbf{X}_{ch}$, $ch = 1,..,Ch$, coding whole control system consists of two parts: part $\mathbf{X}_{ch}^{\text{par}}$ coding parameters and part $\mathbf{X}_{ch}^{\text{red}}$ coding structure.

- **Chromosome Selection.** Chromosome selection can be implemented using a method known from the literature (see e.g. [6], [21]). In our simulations, which results are presented in next section, roulette wheel selection was used. The idea of roulette wheel selection is to promote chromosomes with beneficial fitness function value.
- **Chromosome Crossover.** In this paper crossover with weighting of the genes values was assumed (see e.g. [6], [21]). In crossover take part only those chromosome pairs (selected in previous step), for which drawn real value from range $[0,1]$ is less than crossover probability p_c. $\mathbf{X}_{ch}^{\text{par}}$ parts crossover is described as follows:

$$\begin{cases} X_{j1,g}^{\text{par}} := (1-\phi) \cdot X_{j1,g}^{\text{par}} + \phi \cdot X_{j2,g}^{\text{par}} \\ X_{j2,g}^{\text{par}} := (1-\phi) \cdot X_{j2,g}^{\text{par}} + \phi \cdot X_{j1,g}^{\text{par}} \end{cases}, \tag{3}$$

where $j1$, $j2$ denote in-pair chromosome index, $g = 1,..,L$, denotes gene index, $\phi \in (0,1)$ denotes trial and error selected algorithm parameter. Of course for $\mathbf{X}_{ch}^{\text{par}}$ crossover can be used other method applicable for real coding. It is important to notice that $\mathbf{X}_{ch}^{\text{red}}$ crossover is the same as in classic genetic algorithm (see e.g. [15]).
- **Chromosome Mutation.** Only those chromosomes take part in mutation, for which drawn real value from range $[0,1]$ is less than mutation probability p_m. $\mathbf{X}_{ch}^{\text{par}}$ mutation is described as follows:

$$X_{ch,g}^{\text{par}} := X_{ch,g}^{\text{par}} + \sigma \cdot \text{random}\{-1,+1\} \cdot (pmax_g - pmin_g), \tag{4}$$

where $\sigma \in (0,1)$ denotes mutation intensity parameter, $pmax_g$, $g = 1,..,L$, denotes highest accepted gene value, $pmin_g$, $g = 1,..,L$, denotes lowest accepted gene value. It is important to notice that $\mathbf{X}_{ch}^{\mathrm{red}}$ mutation is the same as in classic genetic algorithm (see e.g. [15].

- **Chromosome Repair.** The aim of the repair is to preserve in search range values of genes that are coding control system parameters. Chromosome repair is described as follows:

$$X_{ch,g}^{\mathrm{par}} := \begin{cases} pmin_g & \text{for} \quad X_{ch,g}^{par} < pmin_g \\ pmax_g & \text{for} \quad X_{ch,g}^{par} > pmax_g \\ X_{ch,g}^{\mathrm{par}} & \text{otherwise} \end{cases}. \tag{5}$$

- **Chromosome Evaluation.** Chromosome evaluation function was set to minimize: RMSE error, zero crossing number of controller output signal, controller output signal dynamics and overshoot of the control signal. High zero crossing number of controller output signal is a negative phenomenon, because it tends to excessive use of mechanical control parts and may cause often huge changes of the controller output signal value. While the overshoot of the control signal is not acceptable in many industrial applications (see e.g. [24]). Chromosome evaluation function is described as follows:

$$ff(\mathbf{X}_{ch}) = (RMSE_{ch} + c_{ch} \cdot w_c + z_{ch} \cdot w_z + ov_{ch} \cdot w_{ov})^{-1}, \tag{6}$$

where $c_{ch} > 0$ denotes the complexity of the controller structure and is calculated by the formula:

$$c_{ch} = \sum_{g=1}^{L} \mathbf{X}_{ch,g}^{\mathrm{red}}, \tag{7}$$

$w_c \in [0,1]$ denotes weight factor for the complexity of the controller structure, $z_{ch} \geq 0$ denotes zero crossing number of controller output signal (in simulations its value is set automatically), $w_z \in [0,1]$ denotes weight for the zero crossing factor, $ov_{ch} \geq 0$ denotes value of the greatest overshoot of the controlled s^1 signal and finally $w_{ov} \in [0,1]$ denotes weight for the overshoot factor. RMSE error function of the ch chromosome is described by the following formula:

$$RMSE_{ch} = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} {\varepsilon_{ch,i}}^2}, \tag{8}$$

where $i = 1, \ldots, N$, denotes sample index, N denotes the number of samples, ε denotes controller tracking error defined as follows:

$$\varepsilon_{ch,i} = s_{ch}^{*} - s_{ch}^{1}, \tag{9}$$

where s^* denotes the value of the reference signal of the controlled value, while s^1 denotes its current value. In our method we maximize the function described in formula (6).

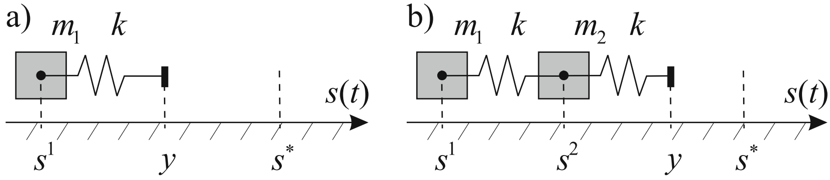

Fig. 3. Simulated spring-mass-damp objects: a) "type 1m", b) "type 2m"

4 Simulations Results

In our simulations there was considered a problem of design controller structure and parameter tuning for two cases: second and fourth-order controlled objects respectively:

- Single spring-mass-damp object, denoted as "type 1m" (Fig. 3.a).
- Double spring-mass-damp object, denoted as "type 2m" (Fig. 3.b).

We took the following assumptions in the simulations:

- Evolutionary algorithm population number was set to 100 chromosomes. Algorithm was executed for 300 iterations (generations), crossover probability value $p_c = 0.8$, mutation probability was $p_m = 0.2$, crossover weight ϕ was random value from range (0,1).
- Fitness function weights were set as follows $w_c = 0.01$, $w_z = 0.001$ and $w_{ov} = 0.001$.
- Simulation length was set to 10 seconds, a shape of the reference signal s^* is presented in Fig. 4 and Fig. 5.
- Search range for genes coding for controller parameter (5) were set as follows: $P_1 = [0,15]$, $I_1 = [0,15]$, $D_1 = [0,1]$, $P_2 = [0,15]$, $I_2 = [0,15]$ and $D_2 = [0,1]$.
- Output signal of the controller was limited to the range $y \in (-2, +2)$.
- Quantization resolution for the output signal y of the controller as well as for the position sensor for s^1 and s^2 was set to 10 bit.
- Time step in the simulation was equal to $T = 0.1$ms, while interval between subsequent controller activations were set to twenty simulation steps.
- The motion equations for position s^1, velocity v^1 and acceleration a^1 for object "type 1m" are described as follows:

$$s_n^1 = s_{n-1}^1 + v_{n-1}^1 \cdot T + (a_{n-1}^1 \cdot T^2) \cdot 0.5, \tag{10}$$

$$v_n^1 = v_{n-1}^1 + a_{n-1}^1 \cdot T, \tag{11}$$

$$a_n^1 = (\left(y_n - s_n^1\right) \cdot k - v_n^1 \cdot \mu) \cdot m_1^{-1}, \tag{12}$$

where n and $n-1$ denotes current and previous simulation step respectively, k is spring constant, y is controller output signal and μ is coefficient of kinetic friction. For object "type 2m" equations for position and velocity of mass

m_1 are identical as in object "type 1m", while the acceleration is described as follows:

$$a_n^1 = \left((s_n^2 - s_n^1\right) \cdot k - v_n^1 \cdot \mu) \cdot m_1^{-1}. \tag{13}$$

Analogically, for mass m_2, the motion equations for position s^2, velocity v^2 and acceleration a^2 have the following form:

$$s_n^2 = s_{n-1}^2 + v_{n-1}^2 \cdot T + (a_{n-1}^2 \cdot T^2) \cdot 0.5, \tag{14}$$

$$v_n^2 = v_{n-1}^2 + a_{n-1}^2 \cdot T, \tag{15}$$

$$a_n^2 = \left((y - s_n^2\right) \cdot k - v_n^2 \cdot \mu) \cdot m_1^{-1}. \tag{16}$$

- Object parameters values were set as follows: spring constant k was set to 10 N/m, coefficient of friction $\mu = 0.5$, masses $m_1 = m_2 = 0.2$ kg. Initial values of: s^1, v^1, s^2 i v^2 were set to zero.
- During simulation three control systems were tested:
  - System with one PID controller (Fig. 2a) denoted as 1xPID-s.
  - Cascaded system combined from two PID controllers (Fig. 2b), where the internal controller was coupled with the signal v^1 denoted as 2xPID-v.
  - Cascaded system combined from two PID controllers (Fig. 2c), where the internal controller was coupled with the signal s^2 (denoted as 2xPID-s).

The idea of the controller structure optimization was to make test of the different controller structures in the same external conditions, calculate the quality factors and choose the best controller.

At the first stage there was investigated a control system whose task was to control the position of the spring-mass-dump object "type 1m" (Fig. 3a). The results are shown in Table 1 in columns named '1xPID+1m' and '2xPID-v+1m' and presented in Fig. 4b and Fig. 4c respectively. In order to better illustrate the control problem in Fig. 4a there was presented the behaviour of the open-loop control system. How we can see, the first controller structure (1xPID) was sufficient to control the second-order object. The use of more complex (cascaded) controller (2xPID-v) was not necessary. The RMSE values for both cases was similar.

In the next stage of the experiment the fourth-order control object "type 2m" was investigated. The results are shown in Table.1 in columns named '1xPID+2m', '2xPID-v+2m','2xPID-s+2m' and presented in Fig.5a-c respectively.

As we can see, the first controller structure (case 1xPID-s+2m) was unable to control this fourth-order object with an sufficient quality (Fig. 5a). The oscillation of the mass position was unacceptable. Similarly, the cascaded controller (2xPID-v) with an internal coupling signal v^1 did not provide a good control quality (Fig. 5b). Only the last controller structure was able to control this object with a good quality (Fig. 5c). The values of quality factors (i.e. RMSE values

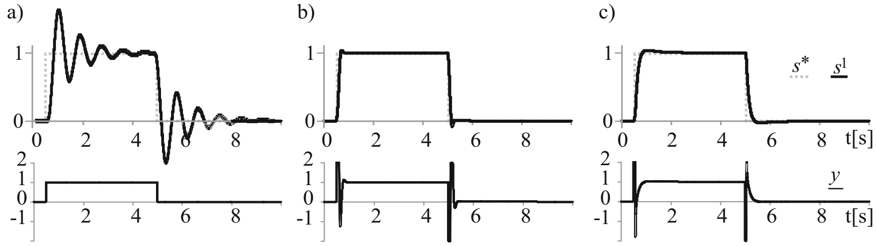

Fig. 4. Graph of the actual signal s^1 and reference value signal s^*, controller output signal y in simulation with the object "type 1m" and for controller: a) open-loop, b) 1xPID-s and c)2xPID-v

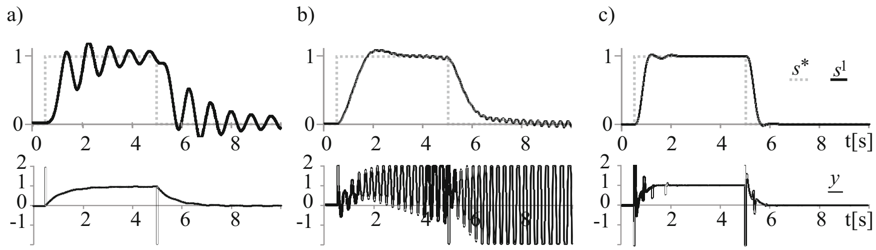

Fig. 5. Graph of the actual signal s^1 and reference value signal s^*, controller output signal y in simulation with the object "type 2m" and for controller: a) 1xPID-s, b) 2xPID-v and c) 2xPID-s

last row in Table 1) confirm the above observations. As a result, this quality factor together with properly defined fitness function (6) can be effectively used for the automatic optimization of the controller with the help of the evolutionary algorithm.

5 Summary

In this paper a new method for optimal controller design was proposed. A characteristic feature of this method is design of both controller parameter and structure using evolutionary algorithm. In order to prove the proposed method, three different controller structures with two different object types were tested. The simulations confirmed correctness of the proposed method.

Further research will be related to developing a general description of a variety of (admissible) control structures. This will allow automation of the design of the optimal controller structure to the specific control object, in the presence of various disturbances and other process control limitations.

Table 1. Parameters of evolutionary designed control systems

Name	Parameters of control systems				
	1xPID+1m	2xPID-v+1m	1xPID-s+2m	2xPID-v+2m	2xPID-s+2m
P_1	15.0	15.0	0.0217	2.06	1.01
I_1	9.63	reduced	1.37	3.08	6.03
D_1	0.7807	0.7850	0.03	0.15	0.20
P_2	-	6.139	-	0.50	1.07
I_2	-	15.0	-	0.08	0.04
D_2	-	reduced	-	0.20	1.60
$RMSE$	0.1209	0.1309	0.3344	0.3102	0.2296
$c_{ch} \cdot w_c$	0.03	0.04	0.03	0.06	0.06
$z_{ch} \cdot w_z$	0.008	0.008	0.008	0.063	0.021
$ov_{ch} \cdot w_o v$	0.0002	0.0001	0.0002	0.0001	0.0000
ff^-1	0.1591	0.179	0.3726	0.4333	0.3106
ff	6.287	5.586	2.684	2.308	3.219

Acknowledgment. The project was financed by the National Science Center on the basis of the decision number DEC-2012/05/B/ST7/02138.

References

1. Cordon, O., Herrera, F., Hoffman, F., Magdalena, L.: Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases. Word Scientific (2001)
2. Cpałka, K.: A New Method for Design and Reduction of Neuro-Fuzzy Classification Systems. IEEE Transactions on Neural Networks 20(4), 701–714 (2009)
3. Cpałka, K.: On evolutionary designing and learning of flexible neuro-fuzzy structures for nonlinear classification. Nonlinear Analysis Series A: Theory, Methods and Applications 71(12), e1659–e1672 (2009)
4. Cpałka, K., Rutkowski, L.: A new method for designing and reduction of neuro-fuzzy systems. In: 2006 IEEE International Conference on Fuzzy Systems, pp. 1851–1857 (2006)
5. Curkovic, M., Jezernik, K., Horvat, R.: FPGA-Based Predictive Sliding Mode Controller of a Three-Phase Inverter. IEEE Transactions on Industrial Electronics 60(2), 637–644 (2013)
6. Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, 3rd edn. IEEE Press, Piscataway (2006)
7. Gabryel, M., Cpałka, K., Rutkowski, L.: Evolutionary strategies for learning of neuro-fuzzy systems. In: I Workshop on Genetic Fuzzy Systems, Genewa, pp. 119–123 (2005)
8. Gabryel, M., Rutkowski, L.: Evolutionary Learning of Mamdani-Type Neuro-fuzzy Systems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 354–359. Springer, Heidelberg (2006)
9. Gabryel, M., Rutkowski, L.: Evolutionary methods for designing neuro-fuzzy modular systems combined by bagging algorithm. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 398–404. Springer, Heidelberg (2008)

10. Hsu, C., Juang, C.: Continuous ant optimized type-2 fuzzy controller for accurate mobile robot wall-following control. In: International Conference on Fuzzy Theory and it's Applications (iFUZZY), 2012, pp. 187–191 (2012)
11. Li, X., Er, M.J., Lim, B.S., et al.: Fuzzy Regression Modeling for Tool Performance Prediction and Degradation Detection. International Journal of Neural Systems 20(5), 405–419 (2010)
12. Kaufman, Y., Ellenbogen, A., Meir, A., Kadmon, Y.: Nonlinear neural network controller for thermal treatment furnaces. In: IEEE 27th Convention of Electrical & Electronics Engineers in Israel (IEEEI), pp. 1–4 (2012)
13. Korytkowski, M., Gabryel, M., Rutkowski, L., Drozda, S.: Evolutionary methods to create interpretable modular system. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 405–413. Springer, Heidelberg (2008)
14. Koza John, R., Streeter Matthew, J., Keane Martin, A.: Routine high-return human-competitive automated problem-solving by means of genetic programming. Information Sciences 178(23), 4434–4452 (2008)
15. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer (1999)
16. Katsuhiko, O.: Modern Control Engineering. Prentice Hall (2001)
17. Przybył, A.: Doctoral dissertation: Adaptive observer of induction motor using artificial neural networks and evolutionary algorithms. Poznan University of Technology (2003) (in polish)
18. Przybył, A., Cpałka, K.: A new method to construct of interpretable models of dynamic systems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS (LNAI), vol. 7268, pp. 697–705. Springer, Heidelberg (2012)
19. Przybył, A., Smolag, J., Kimla, P.: Real-time Ethernet based, distributed control system for the CNC machine. Electrical Review 2010-2 (2010) (in polish)
20. Rasoanarivo, I., Brechet, S., Battiston, A., Nahid-Mobarakeh, B.: Behavioral Analysis of a Boost Converter with High Performance Source Filter and a Fractional-Order PID Controller. In: IEEE Industry Applications Society Annual Meeting (IAS), pp. 1–6 (2012)
21. Rutkowski, L.: Computational Intelligence. Springer (2007)
22. Rutkowski, L., Cpałka, K.: Flexible weighted neuro-fuzzy systems. In: Proceedings of the 9th Neural Information Processing, pp. 1857–1861 (2002)
23. Rutkowski, L., Przybył, A., Cpałka, K., Joo, E.M.: Online Speed Profile Generation for Industrial Machine Tool Based on Neuro-Fuzzy Approach. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part I. LNCS, vol. 6113, pp. 645–650. Springer, Heidelberg (2010)
24. Rutkowski, L., Przybył, A., Cpałka, K.: Novel Online Speed Profile Generation for Industrial Machine Tool Based on Flexible Neuro-Fuzzy Approximation. IEEE Transactions on Industrial Electronics 59(2), 1238–1247 (2012)
25. Teja Ravi, A.V., Chakraborty, C., Maiti, S., Hori, Y.: A New Model Reference Adaptive Controller for Four Quadrant Vector Controlled Induction Motor Drives. IEEE Transactions on Industrial Electronics 59(10), 3757–3767 (2012)